Deducing Interactions in Partially Unspecified Biological Systems

Paolo Baldan[1], Andrea Bracciali[2], Linda Brodo[3], and Roberto Bruni[2]

[1] Dipartimento di Matematica Pura e Applicata, Università di Padova, Italia.
[2] Dipartimento di Informatica, Università di Pisa, Italia.
[3] Dipartimento di Scienze dei Linguaggi, Università di Sassari, Italia.

baldan@math.unipd.it, {braccia,bruni}@di.unipi.it, brodo@uniss.it

Abstract. We show how a symbolic approach to the semantics of process algebras can be fruitfully applied to the modeling and analysis of partially unspecified biological systems, i.e., systems whose components are not fully known, cannot be described entirely, or whose functioning is not completely understood. This adds a novel deductive perspective to the use of process algebras within systems biology: the investigation of the behavioural or structural properties that unspecified components must satisfy to interact within the system. These can be computationally inferred, extending the effectiveness of the in silico experiments. The use of the approach is illustrated by means of case studies.

1 Introduction and Motivations

The convergence of mathematical, technical and natural sciences yields multidisciplinary approaches that can help in better understanding biological phenomena. The formal modeling of such phenomena has recently gained a lot of attention, see e.g. [31,34,20,21,17]. Among these approaches, *process algebras* provide expressive descriptions, enjoy friendly syntax, compositionality and generally support software simulation. To some extent, they appear as easily accessible formalisms, particularly suited for such interdisciplinary research, that favor cross-fertilisation between the two fields: existing calculi have been sometimes applied rather directly, like in the case of stochastic semantics for the Pi-calculus [27,29], while in other cases new language primitives have been specifically designed to capture molecular and biological interaction, like the explicit treatment of membrane nesting [30], membrane activity [7], probability-based reactions [25], active sites in a protein [11] and structure-determined reactions [28,12]. These linguistic abstractions are generally complemented with suitable formal semantics that may describe system behaviour both qualitatively and quantitatively, e.g., in terms of happening reactions and their dynamic constants (i.e., stochastic semantics [26,32] based on Gillespie's algorithm [16]). Often, executable counterparts are provided so that system properties can be both assessed theoretically and verified by means of *in silico* simulations. Encouraging results, e.g. in terms of the coherence between *in silico* and *in vitro* experiments, have been obtained [22,4,10].

This work has been partially supported by the MIUR project *Bisca*.

Within this line of research, that exploits the analogy between biological and software systems, we address the problem of dealing with qualitative analysis of *open systems*. In the context of computer science open systems account for components that are not fully specified or may dynamically join the system at a later stage, such as applications that access services on the network, or proprietary software components. In the biological setting open systems may play the part of not fully understood cellular and chemical compounds.

Here we apply *symbolic transition systems* [2,3], originally developed for software open systems, to the modeling and analysis of biological systems. We aim to show that: *(i)* the symbolic model adds a deductive dimension to the *in silico* experiments, allowing one *to derive the (most general) features that unknown components exhibit when interacting within a system*, and *(ii)* the framework is *language-independent*, in the sense that it applies to a variety of different modeling problems at different levels of abstraction.

The main ingredients of the approach are an algebraic syntax, an operational semantics in terms of suitable labelled transitions and the possibility to deal uniformly with open and closed systems. The approach will be exemplified on a few case studies, related to different levels of abstractions, granularity, and aspects of interest.

Next, we give an informal account of the modeling of a small scenario comprising a virus v and a cell c. Take the (closed) system

$$E \;=\; v[in\ c.\ rna] \mid c[open\ v.\ (prot \mid rna^{\perp})] \tag{1}$$

On the biological side, terms like $v[\dots]$ can be understood as membraned components, while action prefixes like *in c* model reaction capabilities. The name of the membrane identifies the kind of the component. For instance, the virus v is ready to sequentially execute the actions *in c* and *rna*, modeling, respectively, the capability to enter a cell of kind c and then to communicate some RNA information. The cell c "reacts" and opens the membrane of v (action *open v*, i.e., the system comprises some sort of location-awareness by means of membrane names).

The evolution of the system is modeled via labeled transitions from one configuration to the next. The labels record events that are visible to an external observer. The special label τ is used when the corresponding event is an internal reaction, transparent to the outside. In our example, the virus can enter the cell, the membrane v is opened and the RNA interaction takes place: the compound $prot \mid rna^{\perp}$ interacts, without further consequences in this example, with the virus RNA by means of the complementary action $rna^{\perp}$ (*prot* information is disregarded by the virus).

$$E \;\rightarrow_{\tau}\; c[v[rna] \mid open\ v.(prot \mid rna^{\perp})] \rightarrow_{\tau} c[rna \mid prot \mid rna^{\perp}] \rightarrow_{\tau} c[prot] \tag{2}$$

The infinite set of transitions relative to all the terms of the calculus can be finitely specified by a set of structured operational semantics (sos) rules. For instance, all the transitions about a membraned component $m[in\ n.Q \mid R]$ entering the membrane $n[P]$, or a component $open\ n.Q$ destroying the membrane of $n[P]$ are respectively modelled by rules (*in*) and (*open*) in Fig. 1, valid for all m, n, P, Q, R. Analogously, if any

$$\frac{}{m[in\ n.Q \mid R] \ \mid\ n[P] \ \rightarrow_\tau\ n[m[Q \mid R] \ \mid P]}\ (in) \qquad \frac{}{n[P] \mid open\ n.Q \ \rightarrow_\tau\ P \mid Q}\ (open)$$

$$\frac{P_1 \ \rightarrow_\alpha\ Q_1 \quad P_2 \ \rightarrow_{\alpha^\perp}\ Q_2}{P_1 \mid P_2 \ \rightarrow_\tau\ Q_1 \mid Q_2}\ (comm)$$

Fig. 1. Rules for membraned components.

two components P_1 and P_2 can exhibit complementary actions α and $\alpha^\perp$, then by rule (*comm*) their reaction generates a τ transition.

Imagine that the content of the virus cannot be fully characterised, e.g., because not fully understood. In this case we regard E as an *environment*: an open biological system modeled as a term with place-holders X, whose unknown components could be disclosed only dynamically (e.g. when they react to certain stimuli) or where *components* (i.e., closed systems) or other sub-environments can be dynamically plugged in. That is

$$E[X] \ = \ v[\ X\ ] \mid c[open\ v.(prot \mid rna^\perp)] \tag{3}$$

One possibility is to study the closures $E[p]$ of $E[X]$ w.r.t. all the possible closed components p. When simulation is attempted *in silico*, then infinitely many p must be considered. Moreover, conceptually, this approach prevents the dynamic disclosure of environments to be considered, since they are fully exposed at the beginning.

Symbolic transition systems (STS) allow environments as states and logic formulae as transition labels. They exploit the idea that the behaviour of $E[X]$ depends on the applicable semantic rules, which can be partly determined by means of the known structure of $E[X]$ itself, and may, in turn, impose a requirement over X in order make the rule applicable. The formulae of STS transitions, which annotate unknown components with their relevant behavioural or structural requirements, can be composed throughout an execution trace of the environment and represent the "inferred" constraints that an unknown component must fulfill to drive the system to a given state. This allows us to attack problems like predicting the environmental conditions that let a virus reproduce. For instance, the open system $E[X]$ can evolve via suitable "abstractions" of the transitions in (2) for the closed E:

$$E[X] \overset{in\ c.Y|Z}{\longrightarrow}_\tau c[v[Y \mid Z] \mid open\ v.(prot \mid rna^\perp)] \overset{Y,Z}{\longrightarrow}_\tau c[Y \mid Z \mid prot \mid rna^\perp] \overset{\diamond rna W,Z}{\longrightarrow}_c [W \mid Z \mid prot] \tag{4}$$

The first one exhibits the formula $in\ c.Y \mid Z$: the unspecified component X should "at least" be able to perform $in\ c$ (hence X must "know" c) and then behave as $Y \mid Z$, as required by rule (*in*). The second one imposes no constraints since the environment evolves autonomously, the third one requires on Y the capability to interact by means of rna.

Composition of formulae is relevant for the analysis of the evolution of partially specified bio-environments. Indeed, the formula $in\ c.(\diamond rna.W \mid Z)$, obtained by composing the formulae along the execution trace, generalises the capabilities required to

X in order to carry on the overall interaction within the environment (and it is satisfied by the component $p = in\ c.rna \equiv in\ c.(rna.0 \mid 0)$ that instantiates $E[X]$ to $E[p] = E$).

Synopsis. In § 2 we recall the basics of STS. The framework for the analysis of bio-processes is illustrated in § 3 by discussing, in two examples, how it can be used to reason with incomplete information. The first example is based on an original formalisation of the life cycle of the λ-phage virus in BioAmbients. The second example deals with a model of viral cell infection, originally from [1] and used in [7] to introduce Brane Calculi. Concluding remarks and future perspectives are in § 4.

2 Symbolic Operational Semantics

This section recalls the key definitions about symbolic transition systems (see [2,3] for a more comprehensive formal presentation).

Definition 1 (Symbolic Transition System). *A* symbolic transition system *(STS) $\mathcal{S}$ is a set of transitions*

$$C[X_1, \ldots, X_n] \xrightarrow{(\varphi_1,\ldots,\varphi_n)}_a D[Y_1, \ldots, Y_m]$$

where $C[\boldsymbol{X}]$ and $D[\boldsymbol{Y}]$ are environments, a an action label and φ_i are formulae over variables $\{Y_1, \ldots, Y_m\}$ (in a suitable logic, as defined below).

Informally, a symbolic transition represents the fact that the environment $C[\boldsymbol{X}]$ can exhibit an action a and evolve to $D[\boldsymbol{Y}]$ whenever the holes $\boldsymbol{X}$ are filled with any components satisfying $\boldsymbol{\varphi}$. The label $\boldsymbol{\varphi}$ should encode the "least necessary" conditions that components should fulfill for properly taking part to the transition. For the sake of this presentation, following [2], we exploit the STS logic SL, defined below, with action and structural modalities in the style of the ambient logic [9]. However, different choices are conceivable, depending on the calculus of interest.

Spatial modalities emerge when, in order to perform a transition, an environment $E[\boldsymbol{X}]$ must match the left-hand side of the conclusion of a rule. This may require a certain structure to the components that may possibly be plugged in, hence requiring the constructors of the calculus, like $\_ \mid \_$ or $n[\_]$, to appear as terms of the logic, which we call *spatial operators*. Furthermore, the premises of the matched rule must be satisfiable. Such premises may typically require each plugged component to be able to exhibit some behaviour, as in rule (*comm*). Hence, the logic also includes modal operators $\diamond a\ \_$ expressing the capability to perform an action a. A formula which does not impose any constraint on the component is represented as a logical variable X, called the *residual placeholder*.

Definition 2 (SL). *The formulae of the* STS logic SL *are*

$$\varphi ::= X \mid \diamond a\, \varphi \mid f(\varphi_1, \ldots, \varphi_n)$$

where X is a residual pleaceholder, a is an action and f is a spatial operator. A component p satisfies *the formula φ, if $p \models \varphi$ holds according to the following rules:*

$$
\begin{array}{ll}
p \models X & \\
p \models f(\varphi_1, \ldots, \varphi_n) & \textit{if } \exists p_1, \ldots, p_n.\ p \equiv f(p_1, \ldots, p_n) \wedge \forall\, i.\ p_i \models \varphi_i \\
p \models \diamond a\, \varphi & \textit{if } \exists q.\ p \rightarrow_a q \ \wedge\ q \models \varphi
\end{array}
$$

For example, the component $p = c^{\perp}.\,0 \mid a.\,b.\,0$ satisfies the formula $\diamond a\, X$, namely $p \models \diamond a\, X$, because $p \rightarrow_a c^{\perp}.\,0 \mid b.\,0$ (since $a.\,b.\,0 \rightarrow_a b.\,0$) and $c^{\perp}.\,0 \mid b.\,0 \models X$.

Given a formula φ and n components $q_1, \ldots, q_n$, we write $\varphi[q_1/X_1, \ldots, q_n/X_n]$ for the formula obtained from φ by replacing variables X_i with components q_i. Analogously, we denote by $\varphi[\varphi_1/X_1, \ldots, \varphi_n/X_n]$ the formula obtained from φ by replacing variables X_i with formulae φ_i. Variables in formulae stand for the residual q of p, after that p has exhibited the capabilities and/or the structure imposed by the formula.

Definition 3 (Satisfaction with residuals). *Let p be a component, $\boldsymbol{q} = q_1, \ldots, q_n$ a tuple of components and $\varphi \in \mathsf{SL}$ a formula whose variables are contained in $\{X_1, \ldots, X_n\}$. Then, we say that p* satisfies φ with residuals $q_1, \ldots, q_n$*, written $p \models \varphi; \boldsymbol{q}$, whenever $p \models \varphi[q_1/X_1, \ldots, q_n/X_n]$.*

For example, if $p = n[a^{\perp}.\,0 \mid a.\,b.\,0]$ and $\varphi = n[\diamond a^{\perp}\, X_1 \mid a.X_2]$, then trivially $p \models \varphi$, and if $q_1 = 0$ and $q_2 = b.\,0$ then $p \models \varphi; (q_1, q_2)$. We shall write $\boldsymbol{p} \models \boldsymbol{\varphi}; \boldsymbol{q}$ where $\boldsymbol{\varphi} = \varphi_1, \ldots, \varphi_k$ and $\boldsymbol{p} = p_1, \ldots, p_k$ are tuples of formulae and components, respectively, obviously meaning that $p_i \models \varphi_i[q_1/X_1, \ldots, q_n/X_n],\ \forall\, i \in \{1, \ldots, k\}$.

The proper correspondence between the transitions of environments and those of their closed instances, i.e. components, is established by suitable *soundness* and *completeness* properties (see [2,3] for their formal definition). Notably, some sort of "standard" sound and complete STS can be derived for a large class of process calculi (whose semantics is given by SOS rules in suitable formats [3]). Such STS can be constructed by means of a unification-based procedure. All the symbolic transitions spawn from $E[X]$ in (3) are sound.

3 Reasoning with Incomplete Information

We apply now the STS framework to two case studies (the Appendix reports the transitions of their closed specifications). The first one is an original formalization of a pattern of protein interaction relative to the λ−phago virus. Starting from an incomplete BioAmbients specification of the system, the behaviour of one of the proteins can be inferred by reasoning symbolically on the dynamics of the system. The second example, split in two parts, consists of the symbolic reading of biological interaction, also used to introduce the Brane Calculi in [7]. Here, pretending that cell reactions to viruses are not fully understood, we infer the same behaviour described in the original example. Moreover, without changing the experiment, we additionally deduce the (known) mechanisms allowing a given protein to block the virus. These examples are aimed at illustrating the applicability of our approach to different levels of abstraction, and its versatility in supporting the right representation language according to the problem at hand.

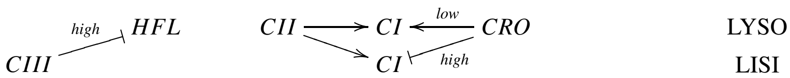

Fig. 2. Hypothesis on inhibition and activation roles of *CRO* and *CIII* proteins.

[VIRUS] = [merge$^+$ virus.([C3] | [C2] | [C1] | [CRO]) | [DNAλ]]

C3 = l_c3!.0 + accept h_c3. pro_c2!.0   C2 = pro_c2?. pro_c1!.0 + enter c2.0

C1 = pro_c1?.(h_cro?. lysi!.0 + l_cro?. lyso!.0)   CRO = l_cro!.0 + h_cro.0

DNAλ = (lyso?.enter dnae.0) + lysi?.($_\lambda$[exit newph.VIRUS] | expel newph)

[ECOLI] = [merge$^-$virus | $_{Dna_e}$[accept dnae] | [HFL]]   **HFL = enter h_c3.0 + X**

Fig. 3. Partial specification of [VIRUS] and [ECOLI]

3.1 Protein Interaction: λ-phage life-cycle

λ-phage simplified life cycle. We consider a simplified representation of the *λ-phage* virus. This virus replicates by binding with the *E.coli* bacterium and injecting its DNA into the bacterium cell. Then, either the virus replicates in several copies until the bacterium membrane is destroyed and the copies released (*lytic pathway*), or the virus DNA merges into the bacterium DNA, the infected bacterium cell multiplies, and its offspring may themselves eventually end up in a lytic pathway (*lysogenic pathway*). The pathway selection is determined by the interaction of the CRO, CI, CII, CIII and HFL proteins in the bacterium cell.

We study the system assuming the following knowledge (see Fig. 2). A high concentration of CI determines the lysogenic cycle, its absence the lytic one. The production of CI is promoted by CII, if it is not inhibited. The role of the bacterial protein HFL is not fully understood, but we know that it can be inhibited by a high concentration of CIII. Moreover, a low concentration of CRO directly stimulates the production of CI, while a high concentration of it destroys CI. Hence, the lysogenic cycle (top row) can be characterised as low CRO and high CIII concentrations, while the lytic one (bottom row) seems to depend on high CRO, exclusively.

BioAmbients uncomplete specification. Under the above hypotheses, the virus and the bacterium can be naturally represented in the BioAmbient calculus as two membraned systems, as shown in Fig. 3 (for a formal description of the BioAmbient calculus we refer the reader to [30]). Proteins are represented as membranes (written [...]) "delimiting" the behaviour they can express. They interact at the same level of nesting: activation is modeled as communication (input/output pairs of actions [..*pro_c*2?..]||[..*pro_c*2!..]) and inhibition as encapsulation ($_1$[..*enter a*..]|| $_2$[..*accept a*..] that evolves in $_2$[..$_1$[..]..]), since this technically blocks the capability of the enclosed protein to communicate in its original environment. The virus consists of the capability to penetrate a suitable membraned environment ([$merge^+$ $virus$..]), i.e., the bacterium cell ([$merge^-$ $virus$..]), and then expose its DNA and express proteins. CIII can either

$$
\begin{aligned}
&{}_{Ecoli}[([\mathbf{C3}] \mid [C2] \mid [C1] \mid [CRO])|[DNA\lambda] \,|_{Dna_e}\, [accept\ dnae] \mid [\mathbf{enterh\_c3.0 + X}]] \\
&\xrightarrow{(l\_c3?.Y_1+Y_2)|Y_3}{}_{Ecoli}\ [{}_{CIII}[0] \mid [\mathbf{C2}] \mid [C1] \mid [CRO])|[DNA\lambda] \mid {}_{Dna_e}[accept\ dnae] \mid {}_{\mathbf{hfl}}[\mathbf{Y_1} \mid \mathbf{Y_3}]] \\
&\xrightarrow{(Y_4+accept\ c2.Y_5|Y_6),Y_3}{}_{Ecoli}\ [({}_{CIII}[0] \mid [C1] \mid [CRO])|[\mathbf{DNA}\lambda] \,|_{Dna_e}\, [accept\ dnae] \mid {}_{\mathbf{hfl}}[{}_{\mathbf{CII}}[\mathbf{0}]|(\mathbf{Y_5} \mid \mathbf{Y_6}) \mid \mathbf{Y_3}]] \\
&\xrightarrow{(Y_7+lysi!.Y_8),Y_6,Y_3}{}_{Ecoli}\ [({}_{CIII}[0] \mid [C1] \mid [CRO])|[{}_{\lambda}[\mathbf{exit\ newph.VIRUS}] \mid \mathbf{expel\ newph}] \mid \\
&\qquad {}_{Dna_e}[accept\ dnae] \,|_{hfl}\, [{}_{CII}[0]|(Y_8 \mid Y_6) \mid Y_3]] \\
&\xrightarrow{Y_8,Y_6,Y_3}{}_{Ecoli}\ [({}_{CIII}[0] \mid [C1] \mid [CRO])|_{\lambda}[\mathbf{VIRUS}] \,|_{Dna_e}\, [accept\ dnae] \,|_{hfl}\, [{}_{CII}[0]|(Y_8 \mid Y_6) \mid Y_3]]
\end{aligned}
$$

Fig. 4. A symbolic trace for ${}_{\lambda}[VIRUS] \mid {}_{\mathrm{Ecoli}}[ECOLI]$.

signal a low concentration or enclose HFL (or any compatible protein) then activating CII by means of a suitable communication. Once activated, CII promotes CI. Moreover, the possibility of CII being itself inhibited has also been modeled (*enter c*2). Sensitivity to high or low concentrations of CRO, modeled by means of suitable communications, causes CI to emit either the lysogenic or the lytic activation signal. This is received by the virus DNA which, accordingly, either enters the bacterium DNA, or expels into the bacterium cell a copy of the virus. The bacterium is modeled as a membrane that can be injected by a virus and contains membraned DNA, which can be accessed by other suitable DNA, and the HFL protein. Importantly, this is represented as a partially specified component, which, as we know, can be inhibited by CIII (*enter h_c*3) but also could alternatively exhibit a behaviour we are not able to specify at the present, represented as variable X.

Symbolic transition system. We study the possible evolutions of the open system ${}_{\lambda}[VIRUS] \mid {}_{\mathrm{Ecoli}}[ECOLI]$ (bio-ambients are sometimes labeled for clarity) in order to understand the possible interactions of HFL within it. In the corresponding STS we can find the trace reported in Fig. 4. (As mentioned in § 2, the labels can be automatically constructed on the basis of the BioAmbient proof rules, while the logic simply consists of the modal operator $\diamond\,\_$, which stands for the possibility of performing an unlabelled transition, and of the spatial operators deriving from the syntax of the calculus). The composition of the formulae over the trace yields an interesting characterisation of the behaviour of HFL:

$$(l\_c3?.(Y_4 + accept\ c2(Y_7 + lysi!.Y_8) \mid Y_6) + Y_2) \mid Y_3$$

It is possible to see that this is a correct abstraction of the actually known HFL (see (10) in Appendix A) and the symbolic trace in Fig. 4 is an abstraction of the corresponding ground trace (see (12)-(16) in Appendix A). Finally, the picture of protein interactions of Fig. 2 can be completed with the relation between low $CIII$, HFL and CII, as shown in Fig. 5, although the partial specification adopted did not have any information about this specific point.

$$
\begin{tikzcd}
 & HFL & CII \arrow[r] \arrow[dr] & CI & CRO \arrow[l, "low"'] \arrow[dl, -|, "high"] & \mathrm{LYSO} \\
CIII \arrow[ur, -|, "high"] \arrow[r, "low"'] & HFL \arrow[r, -|] & CII & CI & & \mathrm{LISI}
\end{tikzcd}
$$

Fig. 5. Full specification of the inhibition and activation schema in Fig. 2.

3.2 Cellular Interaction: Membrane Trepassing

We model an abstraction of a virus replicating its RNA by exploiting a host cell (this has been more exhaustively treated in [7]). The virus membrane complex contains the *capsid*, another membrane complex, which encloses the *nucleocapsid*, i.e. the cytoplasme containing the viral RNA. Here we model the endocytic pathway: the virus penetrates the cell membrane.

We assume that the behaviour of the virus membrane is not known and we deduce it from the operational rules describing the behaviour of the cell. We use fBC, a simplified version of the Brane Calculi, which more suitably models this example. Indeed, fBC focuses on membrane interactions, within the scope of this section, and molecular interactions in the next one. Brane Calculi are intended to model biological interactions inspired by endocytosis/exocytosis, indicated in [7] as *bitonal interactions,* since, informally speaking, they preserve a periodicity between inner and outer areas of membranes.

The calculus fBC (see Fig. 6) can be understood as an extension of BioAmbients, where membranes exhibit themselves a behaviour. The basic membrane complex $\sigma[P]$ consists of an active external membrane layer σ and of complex P inside the membrane ($\diamond$ is the null membrane complex). Other complexes can be obtained by the composition of P and Q, written $P \circ Q$, or as a multiset of molecules, $m_1 \circ \cdots \circ m_k$. Interaction between membrane complexes happens through the active membrane layer σ, which can be halted 0, an action prefixed to an active layer $a.\sigma$ and the parallel composition of active layers $\sigma|\tau$. Membranes behave as follows. $\sigma[P]$ can enter $\tau[Q]$, if σ can execute a p_n action and τ the corresponding coaction $p_n^{\perp}(\rho)$ (with the same n) and $\sigma[P]$ is enclosed within the active membrane ρ, according to the spirit of bitonal reactions (*phago*). In $\tau[\sigma[P]\circ Q]$ the subsystem P can leave the $\tau[\dots]$ membrane complex if σ and τ are ready to execute, respectively, e_n and $e_n^{\perp}$ (*exo*). Finally, $\sigma[P]$ and $\tau[Q]$ merge in $\sigma|\tau[P \circ Q]$ if the membranes can execute m_n and $m_n^{\perp}$ respectively (*mate*).

fBC formalisation. Via a phagocytosis the virus enters the cell wrapped by a membrane. Then, the external membrane of the virus merges with a component of the cell, the *endosome*. Finally, through an exocytosis, the viral *nucleocapsid*, and the viral RNA it contains, is released directly in the *cytosol* of the cell (a possible formalisation in fBC is reported in Appendix B).

Let us suppose now that the mechanisms in the virus membrane are not very well understood. We represent this with the following partial specification, having variable Y in place of the virus membrane.

$$P, Q \;::=\; \diamond \mid \sigma[P] \mid P \circ Q \mid r \qquad \sigma,\rho,\tau \;::=\; 0 \mid a.\tau \mid \sigma|\tau \mid \ldots$$

$$r, s \;::= \diamond \mid m \circ r \quad a \;::= p_n,\ p_n^{\perp}(\sigma),\ e_n,\ e_n^{\perp}, m_n,\ m_n^{\perp} \ldots$$

$$\frac{}{p_n.\sigma \mid \sigma_0\ [P] \ \circ\ p_n^{\perp}(\rho).\tau \mid \tau_0\ [Q] \ \rightarrow\ \tau \mid \tau_0\ [\rho\ [\sigma\ |\sigma_0\ [P]] \ \circ\ Q]}\ (phago)$$

$$\frac{}{e_n^{\perp}.\tau \mid \tau_0\ [e_n.\sigma \mid \sigma_0\ [P] \ \circ\ Q] \ \rightarrow\ P \ \circ\ \sigma \mid \sigma_0 \mid \tau \mid \tau_0\ [Q]}\ (exo)$$

$$\frac{}{m_n.\sigma|\sigma_0[P] \circ m_n^{\perp}.\tau|\tau_0[Q] \rightarrow \sigma|\sigma_0|\tau|\tau_0[P \circ Q]}\ (mate) \qquad \frac{P\ \rightarrow\ Q}{P \circ R\ \rightarrow\ Q \circ R}\ (par)$$

$$\frac{}{r_1 \circ r_1(r_2) \rightrightarrows s_1(s_2).\sigma \mid \sigma_0[r_2 \circ P] \ \rightarrow\ s_1 \circ \sigma \mid \sigma_0[s_2 \circ P]}\ (b\&r) \qquad \frac{P\ \rightarrow\ Q}{\sigma\ [P]\ \rightarrow\ \sigma\ [Q]}\ (mem)$$

Fig. 6. Syntax and operational semantics of fBC.

$$\varphi ::= X \mid \psi[\varphi] \mid \varphi \circ \varphi \mid \diamond_{-}\varphi \mid \theta \quad \text{(complexes)} \qquad \psi \;::= Y \mid \psi|\psi \mid \alpha.\psi \ \text{(membranes)}$$
$$\alpha ::= p_n \mid p_n^{\perp}(\psi) \mid e_n \mid e_n^{\perp} \mid ; \theta(\theta) \rightrightarrows \theta(\theta) \ \text{(actions)} \qquad \theta ::= Z \mid m \mid \theta|\theta \ \text{(molecules)}$$

Fig. 7. The logic associated to the calculus fBC.

$$\textbf{virus} = \mathbf{Y}[\ \textbf{nucap}\ ] \qquad nucap = capsid\ [\ vRNA\ ] \qquad capsid = p_b \mid bud \mid disasm$$
$$cell = p_a^{\perp}(m_a) \mid e_b^{\perp}\ [\ cytosol\ ] \qquad cytosol = endosome\ \circ\ CC \qquad endosome = m_a^{\perp} \mid e_a^{\perp}\ [\ \diamond\ ]$$

The virus content *nucap* is known and it will take part to later stages. It consists of a membrane complex, which contains the RNA and whose active part, *capsid*, is ready to execute a phago action p_b, a *disasm* set of actions that will be defined later, and a *bud* action that is not relevant here. The cell membrane is ready for a phago $p_a^{\perp}(m_a)$ and an exo $e_b^{\perp}$ action (for the reproduced virus eventually leaving the cell). Its content, *cytosol*, consists of a part, denoted *CC*, here not relevant, and the *endosome*, i.e. a membrane complex that can merge $m_a^{\perp}$ with what has been phago-ed and it can uncoat its content $e_a^{\perp}$, in case a suitable coaction can be provided, possibly by the virus.

Symbolic transition system. Also in this example, the associated logic is straightforwardly induced by the syntax of fBC, as shown in Fig. 7. Note that, being fBC semantics unlabelled, the modality $\diamond_{-}\ \psi$ simply stands for the capability of executing any action (e.g., as it may be required by the (*mem*) rule, see Fig. 6). Then, we study the environment:

$$F[Y] = Y[\ nucap\ ]\ \circ\ p_a^{\perp}(m_a)|e_b^{\perp}[\ cytosol\ ]$$

where Y stands for the unknown virus membrane. A possible symbolic trace of the STS of $F[X]$ is:

$$F[Y] \xrightarrow{p_a.Y_1|Y_2} e_b^{\perp}[\ m_a[Y_1|Y_2[nucap]]\ \circ m_a^{\perp} \mid e_a^{\perp}\ [\ \diamond\ ]\ \circ CC\ ]] \tag{5}$$

$$\xrightarrow{Y_1,Y_2} e_b^{\perp}\ [\ e_a^{\perp}\ [\ Y_1|Y_2[\ nucap\ ]]\ \circ\ CC] \tag{6}$$

$$\xrightarrow{e_a.Y_3|Y_4,Y_2} e_b^{\perp}[\ Y_3|Y_4|Y_2[\diamond]\ \circ nucap\ \circ CC] \tag{7}$$

The first symbolic transition (5) constrains the virus membrane to be able to perform a phago p_a action in order to enter the cell via endocytosis, $Y = p_a.Y_1 \mid Y_2$. The requirement is specific for the action offered by the cell membrane. The second symbolic transition (6) does not involve the virus membrane, since the *nucap* of the virus can merge with the *cytosol* of the cell without imposing any further condition on the viral membrane. The formula hence reverts to identity Y_1, Y_2: no requirements over the unspecified components, since the rest of the system is able to evolve autonomously. The last transition (7) requires $e_a.Y_3|Y_4, Y_2$, i.e. the (current state of the) virus membrane should be able to exhibit an action e_a in order to uncoat its content *nucap* via exocytosis. The constraint $Y_1 = e_a.Y_3|Y_4$ is the most general, coherently with the semantic rules.

Inferred information about the unknown components, when they contribute to the overall system behaviour, can be gathered by composing the logical formulae used as labels: any virus whose membrane satisfies $\psi_{\text{STS}} = p_a.\ (e_a.Y_3|Y_4)|\ Y_2$ will be able to enter in the cell and release its *nucap*. Note that ψ_{STS} characterises a general class of components which allow for the interaction of interest. For instance, not only the membrane of the virus $p_a.e_a$ [*nucap*] but also the membranes $p_a.(e_a \mid p_a.e_a)$ or $m_n \mid p_a.e_a$ satisfy ψ_{STS} and, once plugged in Y, are sufficient to drive the system through the same kind of behaviour (and maybe others).

3.3 Biochemical Interaction: Viral RNA Replication

Once inside the cell, the virus capsid is removed (uncoating process) and the virus RNA replicates. Besides this process, also discussed in [7], we address virus neutralisation by the cell. Here, we assume that the mechanisms in the cell content are not fully understood and we show how information about the cell content, relevant for virus replication or neutralisation, can be inferred.

The fBC calculus, in order to express biochemical phenomena, includes a *bind and release* action (*b&r*) to let membranes interact with molecules. The action, denoted $r_1(r_2) \rightrightarrows s_1(s_2)$, can be executed by a membrane if the molecules r_2 are contained in the membrane complex and the molecules r_1 are present outside it. Its effect is to substitute the molecules r_1 with the molecules s_1 outside, and r_2 with the s_2 inside.

We study again a partially specified cell, and then we show how the inferred constraints are coherent with actual components that can reasonably play the part of the unspecified ones.

Virus replication. We assume that the virus $p_a.e_a$ [$nucap$], which fulfills the characterisation inferred in § 3.2, has entered the cell, reaching the state $e_b^\perp[\ p_b|bud|disasm[vRNA]\ \circ CC]$ (a coherent instance of the one in (7)). The action *disasm*, responsible of uncoating the *vRNA*, is specified as a *b&r* action activated by the presence of an outer trigger. It moves the inner *vRNA* outside: $disasm = disTrg(vRNA) \rightrightarrows vRNA(\diamond)$. Moreover, we suppose that the remaining cell content CC is not fully understood, i.e. we focus on the environment

$$G[X] = e_b^\perp[p_b \mid bud \mid disTrg(vRNA) \rightrightarrows vRNA(\diamond)[vRNA] \circ X]$$

A possible symbolic trace of $G[X]$ is the following:

$$G[X] \xrightarrow{disTrg\ \circ\ X_0} e_b^\perp[p_b \mid bud[\diamond] \circ vRNA \circ X_0] \xrightarrow{\xi} e_b^\perp[p_b|bud[\diamond] \circ Z_2 \circ Y_4|Y_5[Z_3 \circ X_6] \circ X_7] \quad (8)$$

The applicable ($b\&r$) rule justifies the first symbolic transition with a spatial constraint requiring that X contains at least a *disTrg* molecule in order to trigger the removal of the viral *capsid*. The second symbolic transition is justified by the *vRNA* molecule, now free within the cell, used as a trigger for another application of ($b\&r$), where $\xi = vRNA(Z_1) \rightrightarrows Z_2(Z_3).\, Y_4|Y_5[Z_1 \circ X_6] \circ X_7$. The formula ξ implies that, triggered by the outer presence of *vRNA*, a set of molecules Z_2 can be released within the cell, so that *vRNA* replication can be supported by the cell. The composition of the formulae in (8) yields

$$\Psi_{\text{STS}} = disTrg \circ vRNA(Z_1) \rightrightarrows Z_2(Z_3).\, Y_4|Y_5[Z_1 \circ X_6] \circ X_7$$

As expected, Ψ_{STS} characterises the mechanisms of virus replication as modeled in [7], which we are following. There, CC is read as providing the suitable triggering and replication (two *vRNA* released) capability:

$$CC = disTrg \circ vRNArepl \circ CC' \quad vRNArepl = vRNA(\diamond) \rightrightarrows vRNA \circ vRNA(\diamond)\ [\diamond]$$

Importantly, the above definition satisfies the characterisation Ψ_{STS} obtained by reasoning symbolically, i.e., $CC \models \Psi_{\text{STS}}$ (where Z_2 stands for $vRNA \circ vRNA$, Z_1, Z_3, X_6, X_7 for $\diamond$, and Y_4, Y_6 for 0). Moreover, the behaviour of $G[CC]$ comprises a trace that is an instance of the symbolic (8), leading, as expected, to a state where *vRNA* has been replicated: $e_b^\perp[\ p_b \mid bud[\diamond] \circ vRNA \circ vRNA \circ CC'\ ]$.

Virus neutralisation. It is interesting to observe another possible evolution of $G[X]$, justified in the corresponding STS by a (*phago*) rule

$$G[X] \xrightarrow{\psi} e_b^\perp[Y_2|Y_3[\ Y_1[\ \ bud \mid disasm[vRNA]\ \ ] \circ X_4]] \quad (9)$$

with $\psi = p_b^\perp(Y_1).Y_2|Y_3[X_4]$. In this case the p_b action of the virus membrane has been exploited to trap the virus *nucap* within a membraned complex ($Y_1[\ ]$).

This evolution, not considered in [7], mimics the presence of a *Mx*-like protein in the cell that inhibits the replication of the virus. This type of proteins seems to play an antiviral activity by trapping the viral capsid and moving it in a location of the cell where the mechanism for the generation of new virus particles becomes unavailable [18]. The simplest representation of the Mx protein can be drawn from ψ as $Mx = p_b^{\perp}(0)[\diamond]$: by means of $p_b^{\perp}(0)$ the viral *nucap* is trapped within an empty membrane. Mx seems to be a coherent simplification of the actual known behaviour of the protein, while $\psi = p_b^{\perp}(Y_1).Y_2|Y_3[X_4]$ characterises the cell components (including the protein *Mx*, but possibly others) capable of trapping the virus within a membrane. Adding the *Mx* protein to a cell with the RNA replication mechanism determines a trace reaching a state where the virus has been phago-ed in a membrane where it can not reproduce (the Mx membrane has been annotated for readability):

$$e_b^{\perp}[\ 0\,\overline{[}\ \ \overline{0}[\ bud \mid disasm[vRNA]\ ]\ \ \overline{]} \circ RC\ ].$$

While the behaviour of Mx proteins has been studied elsewhere, it is worth noting that here it has been inferred from the general rules defining the calculus and an incomplete initial specification. The same specification has led to the inference of the virus replication mechanism. This experiment shows how symbolically reasoning appears as a deductive mechanism, suitable to infer unknown information.

3.4 Discussion

We have presented two proof of concept examples of the application of STS to biological problems. The former is an original formulation of biological interaction that highlights the problem of understanding the interplay of a protein network. The latter is a paradigmatic example of the interaction between a virus and its host cell. These examples have illustrated how, reasoning in presence of incomplete specifications, it has been possible to deduce new knowledge about the studied systems, like

- the emergence of unspecified interactions between bio-components, e.g. the possibility of a HFL-like protein to determine the lytic or lysogenic cycle in the $\lambda-$phage virus life-cycle;
- the constraints over the behaviour and the structure of bio-components needed to participate to the evolution of a system, e.g. the need of the cell content to provide a trigger for *vRNA* replication;
- the discovery of possible components or behaviour not explicitly foreseen in the initial specification of the system under analysis, e.g. the existence and behaviour of an Mx-like protein blocking virus replication.

This poses the problem of characterising the (reachability of) relevant states and evolution traces of the partially specified bio-environments, and, analogously, of proving bio-system properties. For instance one might want to exploit the synthesized STS for the automated state-space exploration in order to characterise "unknown" dangerous bioagents that can compromise the regular activity of a cellular system. An interesting approach in this sense is the definition of a modal logic and a model checking algorithm for Brane Calculi [24] (along the line of Ambient Logic [9]), which defines spatial and temporal properties on membrane systems. Similarly, [13,14,35] exploit Pathway logic and matching algorithms for model checking the evolutions of biological systems. However, beyond similarities, e.g. formulas as labels and unification/matching algorithms, our theoretical framework poses the problem of model checking *open-ended* systems. This is an interesting problem under investigation, whose scope is beyond this paper.

Formulae relative to traces play the part of (minimal) necessary conditions that unspecified components must fulfill to *possibly* drive the system through the trace. Trivially, if $p \models \phi$ and $q \models \psi$ then $p + q$ satisfies both of them and can lead the system through possibly completely unrelated evolutions. This suggests, in general, the difficulties in associating processes to desired behaviours. Besides, while $p + q$ is definitely a process in abstract terms, it might not be feasible in biological terms, so that process characterisation could also require domain specific solutions. This issue is under investigation.

4 Concluding Remarks and Future Work

We have proposed the application of a symbolic approach to the modeling of open biological processes, where some components or features are unknown. An open biological system is seen as a partially specified process of a given calculus tailored to biological processes. Its semantics is given in terms of symbolic transition systems (sts), whose transitions are labeled with logical formulae that express the structural and/or behavioural requirements over the unknown components that let the system evolve. sts can be effectively generated from the sos rules of the process calculus by using a unification-based approach, supporting *in silico* analysis of complex biological systems. Overall, this provides a formal and computational framework capable to infer information about components that are not fully understood beforehand, and that permits the choice of the more appropriate representation language.

To the best of our knowledge, our open-ended and inferential modeling is original in the context of bio process algebras. Indeed, in the literature, simulations and analysis have been carried out starting from completely specified models. Hence, no further information about the behaviour of the system can be inferred, in the sense we do it. A related approach, but in a different perspective, can be [6], where temporal logic is taken as a specification language and machine learning techniques are used to revise the reaction rules initially (fully) available. Another way to deal with incomplete information are discrete approximations, as claimed in [5], but in the significantly different context of the numerical approaches. An analogous unification-based semantics construction is given in [36], in the different context of model-checking for nominal calculi, where unknown can be the communication network.

Beyond the discussed ongoing extensions, a challenging major direction to extend our approach is the use of quantitative and stochastic information, e.g. probabilities and rate constants of reactions, as in [27,29].

Acknowledgments: We are grateful to Pierpaolo Degano and Corrado Priami whose comments have helped us in improving the content and presentation of this paper.

References

1. B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular Biology of the Cell*. Garland, 1994.
2. P. Baldan, A. Bracciali, and R. Bruni. Bisimulation by unification. *Proc. AMAST'02*, *LNCS* 2422, 254–270. Springer, 2002.
3. P. Baldan, A. Bracciali, and R. Bruni. Symbolic equivalences for open systems. *Proc. GC'04*, *LNCS* 3267, 1–17. Springer, 2005.
4. M. Calder, J. Hilston, and S. Gilmore. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. *Trans. on Comp. Systems Biology VII*, *LNBI* 4230, 1–23. Springer, 2006.
5. M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using the PRISM model checker. *Trans. on Comp. Systems Biology VI*, *LNBI* 4220, 179–190. Springer, 2005.
6. L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine learning biochemical networks from temporal logic properties. *Trans. on Comp. Systems Biology VI*, *LNBI* 4220, 68–94. Springer, 2006.
7. L. Cardelli. Brane calculi-interactions of biological membranes. *Proc. CMSB'04*, *LNCS* 3082, 257–280. Springer, 2005.
8. L. Cardelli and A. Gordon. Mobile ambients. *Th. Comp. Sci.*, 240(1):177–213, 2000.
9. L. Cardelli and A. Gordon. Anytime, anywhere. Modal logics for mobile ambients. *Proc. POPL'00*, 365–377. ACM, 2000.
10. D. Chiarugi, M. Curti, P. Degano, and R. Marangoni. VICE: A VIrtual CEll. *Proc. CMSB'04*, *LNCS* 3082, 207–220. Springer, 2004.
11. V. Danos and C. Laneve. Formal molecular biology. *Th. Comp. Sci.*, 325(1):69–110, 2004.
12. P. Degano, D. Prandi, C. Priami, and P. Quaglia. Beta-binders for biological quantitative experiments. *Proc. QAPL'06*, *ENTCS* 164(3), 101–117. Elsevier, 2006.
13. S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway Logic: Symbolic analysis of biological signaling. *Proc. Pacific Symp. on Biocomputing*: 400-412, 2002
14. S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott. Pathway Logic: Executable models of biological networks. *Proc. Rewriting Logic and its Applications*, *ENCS*, 71, 2002.
15. F. Gadducci and U. Montanari. The Tile Model. *Proof, Language and Interaction: Essays in Honour of Robin Milner*, 133–166. MIT Press, 1998.
16. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81(25):2340–2361, 1977.
17. B. Gutnik, D. Pinto, and B. Ermentrout. Mathematical neuroscience: from neurons to circuits to systems. *J. of Physiology*, 97:209–219, 2003.
18. O. Haller and G. Kochs. Interferon-induced Mx proteins: Dynamin-like GTPases with antiviral activity. *Traffic*, 3:710–717, 2002.
19. M. Hennessy and H. Lin. Symbolic bisimulations. *Th. Comp. Sci.*, 138:353–389, 1995.
20. H. de Jong. Modeling and simulation of genetic regulatory systems: a literature review. *J. of Comp. Biology*, 9(1):67–103, 2002.
21. H. Kitano. Systems Biology: a brief overview. *Science*, 295(5560):1662-1664, 2002.
22. P. Lecca, C. Priami, P. Quaglia, B. Rossi, C. Laudanna, and G. Costantin. A stochastic process algebra approach to simulation of autoreactive lymphocyte recruitment. *SIMULATION*, 80(4):273–288, 2004.
23. J. Leifer and R. Milner. Deriving bisimulation congruences for reactive systems. *Proc. CONCUR'00*, *LNCS* 1877, 243–258. Springer, 2000.
24. M. Miculan and G. Bacci. Modal logics for Brane Calculus. *Proc. CMSB'06*, *LNCS* 4210, 1–16. Springer, 2006.
25. M. Nagasaki, S. Onami, S. Miyano, and H. Kitano. Bio-calculus: its concept and molecular interaction. *Genome Informatics*, 10:133–143, 1999.
26. A. Phillips and L. Cardelli. A correct abstract machine for the stochastic pi-calculus. *Proc. Bioconcur'04*, 2004.

27. C. Priami. Stochastic π-calculus. *The Computer Journal*, 38(6): 578-589, 1995.
28. C. Priami and P. Quaglia. Beta-binders for biological interactions. *Proc. CMSB'04*, *LNCS* 3082, 21–34, Springer, 2005.
29. C. Priami and A. Regev and E. Shapiro and W. Silvermann. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80: 25-31, 2001.
30. A. Regev, E. Panina, W. Silverman, L. Cardelli, and E. Shapiro. Bioambients: An abstraction for biological compartements. *Th. Comp. Sci.*, 325(1):141–167, 2004.
31. A. Regev, and E. Shapiro. Cellular Abstractions: Cells as Computation. *Nature*, 419, 343, 2002.
32. A. Regev, W. Silverman, and E. Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. *Proc. Pacific Symp. on Biocomputing*, 459–470, 2001.
33. A. Rensink. Bisimilarity of open terms. *Inform. and Comput.*, 156(1-2):345–385, 2000.
34. C. Surridge. Computational biology. *Nature Insight*, 420(6912):206–246, 2002.
35. C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. *Proc. Pacific Symp. on Biocomputing*, 2004.
36. P. Yang, C. Ramakrishnan, and S. Smolka. A logical encoding of the pi-calculus: Model checking mobile processes using tabled resolution. *Proc. VMCAI'03*, *LNCS* 2575, 86–101. Springer, 2003.
37. E. L. Winnacker. From Genes to Clones *VCH Publishers*, 1987.

A λ-phage life cycle

We report three traces, starting from the completely specified model of the λ-phage life cycle. They express three pathways illustrated in [37], which lead to the lysogenic and lytic cycles. We assume the behaviour of HFL to be:

$$HFL = enterh\_c3.0 \ + \ l\_c3?.acceptc2.lysi!.0 \tag{10}$$

All executions are preceded by the initial phase of virus injection: the two reductions in (11) concerning the merging of the virus and the bacterium membranes. For readability we use the abbreviation: $DNA_e = accept\ dnae.0$.

$${}_{\lambda}[VIRUS] \,|_{Ecoli}\, [ECOLI] \rightarrow_{Ecoli} [([C3] \mid [C2] \mid [C1] \mid [CRO]) \| [DNA\lambda] \mid [DNA_e] \mid [HFL]] \tag{11}$$

In case of a low concentration of CIII the simulation starts at (12):

$${}_{Ecoli}[([\mathbf{C3}] \mid [C2] \mid [C1] \mid [CRO]) \| [DNA\lambda] \mid [DNA_e] \mid [\mathbf{HFL}]] \tag{12}$$

$$\rightarrow_{Ecoli} [{}_{CIII}[0] \mid [\mathbf{C2}] \mid [C1] \mid [CRO]) \| [DNA\lambda] \mid \ [DNA_e] \mid \ {}_{\mathbf{hfl}}[\mathbf{accept\ c2.lysi!.0}]] \tag{13}$$

$$\rightarrow_{Ecoli} [([0] \mid [C1] \mid [CRO]) \mid [\mathbf{DNA}\lambda] \mid [DNA_e] \mid {}_{\mathbf{hfl}}[{}_{\mathbf{CII}}[\mathbf{0}] \| \mathbf{lysi!.0}]] \tag{14}$$

$$\rightarrow_{Ecoli} [([0] \mid [C1] \mid [CRO]) \| [{}_{\lambda}[\mathbf{exit\ newph.VIRUS}] \mid \mathbf{expel\ newph}] \mid [DNA_e] \,|_{hfl}\, [{}_{CII}[0] \| 0]] \tag{15}$$

$$\rightarrow_{Ecoli} [([0] \mid [C1] \mid [CRO]) |_{\lambda} [\mathbf{VIRUS}] \mid [DNA_e] \,|_{hfl}\, [\ {}_{CII}[0] \| 0]] \tag{16}$$

A high concentration of CIII and a low of CRO leads the system to lysogeny, the simulation starts at (17):

$$ {}_{Ecoli}[([\mathbf{C3}] \mid [C2] \mid [C1] \mid [CRO]) | [DNA\lambda] \mid [DNA_e] \mid [\mathbf{HFL}]] \tag{17} $$

$$ \rightarrow_{Ecoli} \; [{}_{\mathbf{CIII}}[\mathbf{pro\_c2}!.\mathbf{0} | \; {}_{\mathbf{HFL}}[]] \mid [\mathbf{C2}] \mid [C1] \mid [CRO]) | [DNA\lambda] \mid \; [DNA_e]] \tag{18} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [\mathbf{pro\_c1}.\mathbf{0}] \mid [\mathbf{C1}] \mid [CRO]) \mid [DNA\lambda] \mid \; [DNA_e]] \tag{19} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [\mathbf{high\_cro}?.\mathbf{lysi}!.\mathbf{0} \; + \; \mathbf{l\_cro}?.\mathbf{lyso}!.\mathbf{0}] \mid [\mathbf{CRO}]) \mid [DNA\lambda] \mid [DNA_e]] \tag{20} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [\mathbf{lyso}!.\mathbf{0}] \mid [0]) \mid [\mathbf{DNA}\lambda] \mid [DNA_e]] \tag{21} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [0] \mid [0]) \mid [\mathbf{enter\ dnae}.\mathbf{0}] \mid [\mathbf{DNA_e}]] \tag{22} $$

$$ \rightarrow_{Ecoli} \; [(\; {}_{CIII}[0| \; {}_{HFL}[0]] \mid \; {}_{CII}[0] \mid \; {}_{CI}[0] \mid \; {}_{CRO}[0]) \parallel \; {}_{Dna_e}[\; {}_{Dna_\lambda}[0]]] \tag{23} $$

Finally, a high concentration of CRO leads the system to lysis, even with a high concentration of CIII, simulation starts at (24), continuing from (20):

$$ {}_{Ecoli}[([0| \; {}_{HFL}[0]] \mid [0] \mid [\mathbf{high\_cro}?.\mathbf{lysi}!.\mathbf{0} \; + \; \mathbf{l\_cro}?.\mathbf{lyso}!.\mathbf{0}] \mid [\mathbf{CRO}]) \mid [DNA\lambda] \mid [DNA_e]] \tag{24} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [\mathbf{lysi}!.\mathbf{0}] \mid [0]) \mid [\mathbf{DNA}\lambda] \mid [DNA_e]] \tag{25} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [0] \mid [0]) \mid [\; {}_{\lambda}[\mathbf{exit\ newph}.\mathbf{VIRUS}] \mid \mathbf{expel\ newph}] \mid [DNA_e]] \tag{26} $$

$$ \rightarrow_{Ecoli} \; [([0| \; {}_{HFL}[0]] \mid [0] \mid [0] \mid [0]) \mid \; {}_{\lambda}[VIRUS] \mid [DNA_e]] \tag{27} $$

B Virus entering a cell

We report the fully specified model of the virus entering a cell.

$$ virus = p_a.e_a \, [\; nucap \;] \qquad nucap = capsid \, [\; vRNA \;] \qquad capsid \; = p_b \mid bud \mid disasm $$

$$ cell \; = p_a^\perp(m_a) \mid e_b^\perp \, [\; cytosol \;] \quad cytosol = endosome \; \circ \; CC \quad endosome = m_a^\perp \mid e_a^\perp \, [\; \diamond \;] $$

The initial configuration of the system gives origin to the following simulation that exhibits the expected behaviour:

$$ p_a.e_a[\; nucap \;] \; \circ \; p_a^\perp(m_a) | e_b^\perp[\; m_a^\perp \mid e_a^\perp \; [\; \diamond \;] \; \circ \; CC \;] \tag{28} $$

$$ \rightarrow e_b^\perp[\; m_a[\; e_a[nucap] \;] \; \circ \; m_a^\perp \mid e_a^\perp \; [\; \diamond \;] \; \circ CC \;] \tag{29} $$

$$ \rightarrow \; e_b^\perp[\; e_a^\perp[e_a[nucap]] \; \circ CC] \; \rightarrow \; e_b^\perp[\; 0[\diamond] \circ \; nucap \; \circ CC] \tag{30} $$